**IN THE CLAIMS:**

Please amend the claims as follows:

1. (Currently Amended) A memory architecture for use with a computing device, comprising:

a compressed memory for storing compressed data;

a compression control register comprising at least block size data; and

a compression engine having a compressor for compressing blocks of uncompressed data from a cache for storage in said compressed memory, wherein a size of the blocks of uncompressed data is defined by the block size data, and a compression monitor for monitoring achieved compression ratios and for providing an indication when the achieved compression ratios fall below a predetermined threshold level, wherein the compression engine changes the block size data in response to the achieved compression ratio falling below the predetermined threshold level.

2. (Original) The memory architecture of claim 1 wherein said compressed memory is divided into a plurality of pages.

3. (Original) The memory architecture of claim 1 wherein said compressed memory includes a setup table comprises a plurality of pointers that link to said pages.

4. (Canceled) The memory architecture of claim 1 wherein said block size data is part of a compression control register.

5. (Previously Presented) The memory architecture of claim 4 wherein, for each page of memory, a corresponding value of the compression control register is stored in a page table.

6. (Original) The memory architecture of claim 1 wherein said compression engine further includes a decompressor for decompressing compressed data to be stored in the cache.

7. (Original) The memory architecture of claim 1, wherein the compression ratio monitor provides an indication by generating a trap to software.

8. (Original) The memory architecture of claim 7, wherein the memory architecture further comprises a compression control register readable by software, wherein the compression control register comprises at least a block size.

9. (Currently Amended) A system, comprising:

a processor for processing data;

a cache for storing, in an uncompressed format, data accessed by the processor;

a compressed memory for storing data accessed by the processor in a compressed format;

a compression engine having a compressor for compressing blocks of uncompressed data from the cache for storage in the compressed memory, wherein the size of the blocks of data that are compressed by said compressor are controlled by block size data, and a compression monitor for generating a software trap when achieved compression ratios fall below a predetermined threshold level; and

an executable component configured to change the block size data in response to detecting the software trap, wherein the executable component is configured to change the block size data by writing to a compression control register accessed by the compression engine.

10. (Original) The system of claim 9, wherein the executable component is a game.

11. (Original) The system of claim 9, wherein the executable component is configured to change the block size data based on one or more system parameters.

12. (Original) The system of claim 11, wherein the one or more system parameters comprises at least one of: one or more currently running tasks and a fill state of said compressed memory.

13. (Canceled) The system of claim 9, wherein the executable component is configured to change the block size data by writing to a compression control register accessed by the compression engine.

14. (Original) The system of claim 9 wherein said compressed memory is divided into a plurality of pages and includes a page table comprising a plurality of pointers that link to said pages.

15. (Original) The system of claim 14 wherein the page table further comprises block size data and a compression ratio corresponding to each page.

16. (Original) The system of claim 9 wherein the software is further configured to:

monitor the available capacity of the compressed memory; and

if the available capacity is below a threshold level, recompress data previously compressed using a first block size, using a second larger block size.

17. (Original) A system, comprising:

a processor for processing data;

a cache for storing, in an uncompressed format, data accessed by the processor;

a compressed memory for storing data accessed by the processor in a compressed format; and

an executable component configured to monitor the available capacity of the compressed memory and, if the available capacity is below a threshold level, recompress data in the compressed memory previously compressed using a first block size, using a second larger block size.

18. (Original) The system of claim 17 wherein said compressed memory is divided into a plurality of pages and includes a page table comprising a plurality of pointers that link to said pages and corresponding block sizes used to compress said pages.

19. (Original) The system of claim 18 wherein the page table further comprises block size data and a compression ratio corresponding to each page.

20. (Original) The system of claim 19, wherein the executable component is configured to periodically interrogate the page table to examine compression ratios for pages stored in the compressed memory.

21. (Original) A method of compressing memory, comprising:

compressing data using a selected data block size to generate compressed blocks of data;

storing the compressed blocks of data in a memory having a physical storage size;

calculating a dynamic compression ratio calculated based on the selected data block size and the size of the compressed blocks; and

increasing the data block size, in response to determining the dynamic compression ratio has fallen below a threshold level.

22. (Original) The method of claim 21, wherein calculating the dynamic compression ratio is performed in hardware.

23. (Original) The method of claim 22, wherein increasing the data block size is performed in software.

24. (Original) The method of claim 23, wherein the software increases the data block size only if the available capacity of the memory is below a predetermined level.

25. (Original) The method of claim 23, wherein the software periodically monitors the available capacity of the memory and, if the available capacity of the memory is below a threshold value, initiates recompression of data previously compressed with a first data block size, using a second larger block size.